IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Daniel ManHung Wong	)	Confirmation No.: 3803
	)	
Serial No.: 10/786,941	)	Examiner: Michael Pham
	)	
Filed on: February 24, 2004	)	Group Art Unit No.: 2167

For: SENDING CONTROL INFORMATION WITH DATABASE STATEMENT

Via EFS-Web
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir or Madam:

    This Appeal Brief is submitted in response to the Notice of Panel Decision from Pre-

Appeal Brief Review mailed on December 3, 2010, and in support of the Notice of Appeal filed

on October 15, 2010.

TABLE OF CONTENTS

I.   REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II.   RELATED APPEALS AND INTERFERENCES

A prior Notice of Appeal was filed on October 29, 2008, for the subject patent

application. As a result of the prior Appeal, the Pre-Appeal Brief Panel decided to reopen

prosecution rather than proceed to the Board of Patent Appeals and Interferences.

A Notice of Appeal was filed on March 5, 2002, for related Application Serial Number

09/167,092. The related Appeal resulted in a Notice of Allowance dated June 18, 2002.

There are no other prior or pending appeals, judicial proceedings or interferences known

to the appellant which may be related to, directly affect or be directly affected by or have a

bearing on the Board's decision in the pending appeal.

III.   STATUS OF CLAIMS

Pending Claims 1-26 have been finally rejected and are the subjects of this appeal.

IV.   STATUS OF AMENDMENTS

Appellants are appealing on the basis of the claims as amended in the reply submitted on

April 28, 2010. The claims were not amended after the Final Office Action of July 21, 2010.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains two independent claims: Claim 1 and Claim 14.

A.   CLAIM 1

Claim 1 features a "machine-implemented method for executing a database statement." See, for example, pars. 4, 21, 23, 25-26, 30, 32-35, and 40; Original Claim 1. The method features "a database server receiving a request to execute the database statement." See, for example, pars. 4, 21, 23, 32-35, and 46-47; Original Claims 1 and 14. The "request includes the database statement and a tag that does not conform to a database language of said database statement." See, for example, pars. 4, 17-18, 20, 22-23, 25, 27, 32-35, 37, 46-47; Original Claims 1 and 14. The "tag is not embedded in said database statement." See, for example, pars. 17 (tags "appended to database statements"), 18 ("binding a tag to each database statement"), 22 (tag "appended to, attached to, sent with, embedded in or otherwise associated with database statement"), 25, 27, 32-35, 37; Title of the Application.

In paragraph 33, the specification provides a particular "example of an execution interface for executing database statements with the tag field" as "DBMS_SQL.PARSE(cursor, statement, language flag, control tag)," with a "second argument" of "statement" and a "final argument" of "control tag." Paragraph 34 explains that "the execution interface may be DBMS_SQL.PARSE(cursor,'SELECT * FROM emp',v7, 'resource=g1 id=scott'), where "the statement being parsed is SELECT * FROM emp," and the "value of the control tag is 'resource=g1 id=scott', which may indicate that the component or application issuing the SQL statement has ID 'g1', and the ID of the user that wrote the database statement is 'scott'."

The tag of Claim 1 "specifies at least one parameter field and at least one parameter value." See, for example, pars. 17, 22-23, 25, 28-30, 32-34, 36-37; Original Claims 1 and 14. Additionally, the method features, "in response to receiving the request, said database server storing information from the tag in a manner that is associated with said database statement and accessible to a tag access mechanism." See, for example, pars. 17-18, 20, 25, 27, 32, 38-47; Original Claims 1 and 14. The method further features "said database server executing said database statement, wherein during execution of said database statement said database server

provides access to one or more of the at least one parameter value through said tag access

mechanism provided by said database server." See, for example, pars. 17-18, 20, 22-23, 25, 27-

30, 32, 34, 39-47; Original Claims 1 and 14.

B.   CLAIM 14

Claim 14 features a "non-transitory machine-readable medium storing one or more

sequences of instructions, which when executed by one or more processors, causes the one or

more processors to perform" the recited steps. See, for example, pars. 25, 38, 40-43, 47;

Original Claim 14. The recited steps include "a database server receiving a request to execute

the database statement." See, for example, pars. 4, 21, 23, 32-35, and 46-47; Original Claims 1

and 14. The "request includes the database statement and a tag that does not conform to a

database language of said database statement." See, for example, pars. 4, 17-18, 20, 22-23, 25,

27, 32-35, 37, 46-47; Original Claims 1 and 14. The "tag is not embedded in said database

statement." See, for example, pars. 17 (tags "appended to database statements"), 18 ("binding a

tag to each database statement"), 22 (tag "appended to, attached to, sent with, embedded in or

otherwise associated with database statement"), 25, 27, 32-35, 37; Title of the Application.

In paragraph 33, the specification provides a particular "example of an execution

interface for executing database statements with the tag field" as "DBMS_SQL.PARSE(cursor,

statement, language flag, control tag)," with a "second argument" of "statement" and a "final

argument" of "control tag." Paragraph 34 explains that "the execution interface may be

DBMS_SQL.PARSE(cursor,'SELECT * FROM emp',v7, 'resource=g1 id=scott'), where "the

statement being parsed is SELECT * FROM emp," and the "value of the control tag is

'resource=g1 id=scott', which may indicate that the component or application issuing the SQL

statement has ID 'g1', and the ID of the user that wrote the database statement is 'scott'."

The tag of Claim 1 "specifies at least one parameter field and at least one parameter value." See, for example, pars. 17, 22-23, 25, 28-30, 32-34, 36-37; Original Claims 1 and 14. Additionally, the recited steps include, "in response to receiving the request, said database server storing information from the tag in a manner that is associated with said database statement and accessible to a tag access mechanism." See, for example, pars. 17-18, 20, 25, 27, 32, 38-47; Original Claims 1 and 14. The recited steps further include "said database server executing said database statement, wherein during execution of said database statement said database server provides access to one or more of the at least one parameter value through said tag access mechanism provided by said database server." See, for example, pars. 17-18, 20, 22-23, 25, 27-30, 32, 34, 39-47; Original Claims 1 and 14.

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A.   Claims 1-26; 35 U.S.C. § 112: Claims 1-26 were rejected under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the written description requirement.

B.   Claims 1-4, 6-9, 14-21, and 26; 35 U.S.C. § 102(e) – *Puz*: Claims 1-4, 6-9, and 14-21 were rejected under 35 U.S.C. 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Patent Application Publication No. 2003/0140311 (*Puz*).

C.   Claims 5, 10-13, and 22-25; 35 U.S.C. § 103(a) – *Puz* and *Fujiwara*: Claims 5, 10-13, and 22-25 were rejected under 35 U.S.C. 35 U.S.C. § 103(a) as allegedly unpatentable in view of U.S. Patent Application Publication No. 2003/0140311 (*Puz*) and further in view of U.S. Patent Application Publication No. 2003/0014394 (*Fujiwara*).

VII.   ARGUMENTS

It is respectfully submitted that the Examiner committed clear error in rejecting Claims 1-26 under 35 U.S.C. § 112, first paragraph, Claims 1-4, 6-9, and 14-21 under 35 U.S.C. 35 U.S.C. § 102(e), and Claims 5, 10-13, and 22-25 under 35 U.S.C. 35 U.S.C. § 103(a).

A.   CLAIMS 1-26; 35 U.S.C. § 112

Claims 1-26 were rejected under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the written description requirement. This rejection was made in clear error.

The written description requirement of 35 U.S.C. § 112, first paragraph, is satisfied when the Appellant conveys, "with reasonable clarity to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention." See *Vas-Cath, Inc. v. Mahurkar*, 935 F.2d 1555, 1563-64, 19 USPQ2d 1111, 1117 (Fed. Cir. 1991).

The Final Office Action of July 21, 2010, erroneously asserted (pg. 2) that there is "no clear support from the specification" for the claimed feature "wherein said tag is not embedded in said database statement." Appellant respectfully submits that the specification provides overwhelmingly clear support for the feature.

The specification as filed describes Claims 1-26 in a way that reasonably conveys to a person of ordinary skill in the art that the inventor had possession of the claimed invention, which includes a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement. The specification (paragraphs 17-18, 22, 25, 27, 32-35, and 37) provide numerous example tags that are (a) included in requests to execute database statements, and (b) not embedded in database statements.

In paragraph 33, The specification provides a particular "example of an execution interface for executing database statements with the tag field." The example execution interface is "DBMS_SQL.PARSE(cursor, statement, language flag, control tag)," with a "second argument" of "statement" and a "final argument" of "control tag." As shown and described, the cursor, the language flag, and the control tag are (a) included in a request to execute a database statement, and (b) not embedded in the database statement.

Paragraph 34 provides an example tag that is (a) included in a request to execute an example database statement, and (b) is not embedded in the example database statement. Paragraph 34 explains that "the execution interface may be DBMS_SQL.PARSE(cursor, 'SELECT * FROM emp', v7, 'resource=g1 id=scott'." The "statement being parsed is SELECT * FROM emp," and "the value of the control tag is 'resource=g1 id=scott', which may indicate that the component or application issuing the SQL statement has ID 'g1', and the ID of the user that wrote the statement is 'scott'." The example control tag is (a) included in a request to execute a database statement, and (b) not embedded in the database statement.

In yet another example, provided in paragraph 37, the tag is sent in an XML statement rather than in a SQL database statement. Paragraph 37 explains that "a tag may be <SQL attributes> <resource> g1 <> <id> scott <> <SQL attributes>." In the example, the tag provides control information such as "g1," the name of the resource issuing the database statement, and "scott," the ID of the developer that wrote the database statement. As shown, the tag is in an XML statement, (b) not embedded in a database statement.

Paragraphs 17-18, 22, 25, and 27 of The specification provide additional support for a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement. For example, paragraph 22 explains that a tag may be "appended to,

attached to, sent with, embedded in or otherwise associated with database statement." The example describes several embodiments where a tag is not embedded in a database statement, including (1) a tag that is "appended to" a database statement (see also par. 17), (2) a tag that is "attached to" a database statement, (3) a tag that is "sent with" a database statement (see also the Title of the subject application), or (4) a tag that is "otherwise associated with database statement." Another embodiment for (5) a tag that is bound to a database statement is described in paragraph 18.

The specification as filed describes Claims 1-26 in a way that reasonably conveys to a person of ordinary skill in the art that the inventor had possession of the claimed invention, which includes a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement. Therefore, the rejection of Claims 1-26 cannot be sustained under 35 U.S.C. § 112, first paragraph. Appellant respectfully requests reversal of the rejection of Claims 1-26 under 35 U.S.C. § 112, first paragraph. Claims 1-26 have been extensively examined and are believed to be in condition for allowance.

B.   CLAIMS 1-4, 6-9, 14-21, AND 26; 35 U.S.C. § 102(e) – *PUZ*

Claims 1-4, 6-9, 14-21, and 26 were rejected under 35 U.S.C. § 102(e) as allegedly anticipated by *Puz*. This rejection was made in clear error.

The Federal Circuit held that "unless a reference discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it cannot be said to prove prior invention of the thing claimed and, thus, cannot anticipate under 35 U.S.C. § 102." *Net MoneyIN, Inc. v. Verisign, Inc.*, Op. No. 2007-1565 (Fed. Cir. 2008).

The Final Office Action of July 21, 2010, erroneously asserted (pgs. 4-5) that the security marker in *Puz* is analogous to the claimed tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, as claimed. The Final Office Action also erroneously asserted (pgs. 4-5) that *Puz* shows a database server that has access to a parameter value of the tag during execution of the database statement, as claimed. *Puz* does not show these features separately, let alone arranged or combined in the same way as recited in Claims 1-4, 6-9, 14-21, and 26. Appellant respectfully submits that the rejection cannot be sustained under 35 U.S.C. § 102(e).

i.   ***Puz* does not show a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement.**

All pending claims recite, in part: "a database server receiving a request to execute the database statement, wherein the request includes the database statement and a tag that does not conform to a database language of said database statement, wherein said tag is not embedded in said database statement." Information from the tag is stored and used during execution of the database statement. The claimed embodiment features a tag that is received with the database statement even though the tag is not embedded in the database statement and does not conform to a database language of the database statement. In other words, the tag is not merely a sub-part of a database statement.

The rejection of Claims 1-4, 6-9, 14-21, and 26 cannot be sustained under 35 U.S.C. § 102(e) at least because *Puz* does not show a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, as claimed. *Puz* relies on the position of a marker within a database statement. The marker of *Puz* would no longer mark

the position if the marker was removed from the database statement. Thus, not only does *Puz* fail to show the claimed feature, but the techniques in *Puz* could not have been modified to produce the claimed feature.

In *Puz*, a security marker is embedded as a placeholder in a database statement (pars. 28, 33). *Puz* states (par. 28) that "an appropriate security marker is inserted into the SQL string." *Puz* repeatedly reinforces the fact that the marker is embedded in the database statement (pars. 28, 29, 30 ("server-side software 12 receives the SQL query parts from the client system 18 as a single string"), 32 ("inserts the aforementioned security markers into the SQL string at appropriate positions"), 33, 35, 36).

Puz relies on the position of the marker within the database statement because the marker is used as a placeholder for another database statement. As explained (par. 33), "each security marker is replaced with respective SQL joins and conditions to form a final SQL string for submission to a DBMS (84)." The server is "delegated the job of interpreting the markers and translating them into SQL" (par. 38). In other words, the security marker is embedded in a first database statement, and the server replaces the security marker with a second database statement. The first database statement and the second database statement are combined to form a final database statement for execution by a database server.

The security marker in *Puz* must be embedded in the first database statement in order to hold a place within the first database statement. *Puz* could not replace the security marker with a second database statement to form a final database statement if the security marker was not embedded in the first database statement. Therefore, not only does *Puz* fail to show a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, as claimed, but *Puz* could not possibly have produced the claimed features.

For at least these reasons, *Puz* does not show the claimed tag, as would be required to sustain a rejection of Claims 1-4, 6-9, 14-21, and 26 under 35 U.S.C. § 102(e). Appellant respectfully requests reversal of the rejection of Claims 1-4, 6-9, 14-21, and 26 under 35 U.S.C. § 102(e).

ii. *Puz* does not show a database server that has access to a parameter value of the tag during execution of the database statement.

All pending claims recite, in part: "wherein said tag specifies at least one parameter field and at least one parameter value; in response to receiving the request, said database server storing information from the tag in a manner that is associated with said database statement and accessible to a tag access mechanism; said database server executing said database statement, wherein during execution of said database statement said database server provides access to one or more of the at least one parameter value through said tag access mechanism provided by said database server."

The rejection of Claims 1-4, 6-9, 14-21, and 26 cannot be sustained under 35 U.S.C. § 102(e) at least because *Puz* does not show a database server that has access to a parameter value of the tag during execution of the database statement, as claimed. As discussed in Section VII.B.1, *Puz* does not show any tags that are sent with but not embedded in the database statement. Thus, *Puz* does not show a database server that has access to a parameter value of the tag during execution of the database statement. Even if the security marker of *Puz* was analogous to the claimed tag, which it is not, *Puz* still fails to show that the database server has access to a parameter value of the tag during execution of the database statement. *Puz* does not use the security marker during execution of the database statement.

Puz (par. 33) replaces the security marker prior to execution of the database statement. In *Puz* (pars. 32-33), a content management system (CMS) receives a SQL string with embedded security markers. In "a second phase of query processing," "each security marker is replaced with respective SQL joins and conditions to form a final SQL string for submission to a DBMS (84)" (pars. 33, 38). *Puz* continues, "The final SQL string is submitted by the server-side CMS to the DBMMS for execution of the final SQL query." In other words, the security marker is replaced before the SQL string is submitted to the database server. The database server in *Puz* does not receive the security marker. Therefore, not only does *Puz* fail to show a database server that has access to a parameter value of the tag during execution of the database statement, as claimed, but the database server in *Puz* could not have possibly accessed any parameter values of the security marker during execution of the database statement.

For at least these reasons, *Puz* does not show a database server that has access to a parameter value of the tag during execution of the database statement, as would be required to sustain a rejection of Claims 1-4, 6-9, 14-21, and 26 under 35 U.S.C. § 102(e). Appellant respectfully requests reversal of the rejection of Claims 1-4, 6-9, 14-21, and 26 under 35 U.S.C. § 102(e).

iii. Conclusion regarding Claims 1-4, 6-9, 14-21, and 26

As discussed above, at least two substantial features of Claims 1-4, 6-9, 14-21, and 26 are completely absent from *Puz*. Even if *Puz* did show the two features separately, which it does not, *Puz* does not show that, during execution of a database statement, a database server is provided with access to a parameter of a tag that is (a) included in a request to execute the database statement, and (b) not embedded in the database statement. Additionally, *Puz* does not

show a database server that has access, during execution, to information from a tag that does not

conform to a database language of the database statement. *Puz* does not contain "all of the

limitations arranged or combined in the same way as recited in the claim," as required by

NetMoneyIN. Therefore, the rejection of Claims 1-4, 6-9, 14-21, and 26 cannot be sustained

under 35 U.S.C. § 102(e). Appellant respectfully requests reversal of the rejection of Claims 1-4,

6-9, 14-21, and 26 under 35 U.S.C. § 102(e). Claims 1-4, 6-9, 14-21, and 26 have been

extensively examined and are believed to be in condition for allowance.

1. CLAIMS 1-4 AND 6-9; 35 U.S.C. § 102(e) – *PUZ*

Claims 2-4 and 6-9 depend from Claim 1 and include each of the limitations of Claim 1.

Claims 2-4 and 6-9 are not argued separately for the purposes of this appeal. The rejection of

Claims 1-4 and 6-9 cannot be sustained under 35 U.S.C. § 102(e).

Claim 1 features a machine-implemented method for executing a database

statement, the method comprising the steps discussed in parts VII.B.i, VII.B.ii, and VII.B.iii,

above. Claim 1 is patentable over the art of record for at least the reasons provided with respect

to these features that are recited in every pending claim. Accordingly, Appellant respectfully

requests reversal of the rejection of Claims 1-4 and 6-9 under 35 U.S.C. § 102(e). Claims 1-4

and 6-9 have been extensively examined and are believed to be in condition for allowance.

2. 14-21 AND 26; 35 U.S.C. § 102(e) – *PUZ*

Claims 15-21 and 26 depend from Claim 14 and include each of the limitations of Claim

14. Claims 15-21 and 26 are not argued separately for the purposes of this appeal. The rejection

of Claims 14-21 and 26 cannot be sustained under 35 U.S.C. § 102(e).

Claim 14 features a non-transitory machine-readable storage medium storing one or more sequences of instructions, which when executed by one or more processors, causes the one or more processors to perform a method comprising the steps discussed in parts VII.B.i, VII.B.ii, and VII.B.iii, above. Claim 14 is patentable over the art of record for at least the reasons provided with respect to these features that are recited in every pending claim. Accordingly, Appellant respectfully requests reversal of the rejection of Claims 14-21 and 26 under 35 U.S.C. § 102(e). Claims 14-21 and 26 have been extensively examined and are believed to be in condition for allowance.

C. CLAIMS 5, 10-13, AND 22-25; 35 U.S.C. § 103(a) – *PUZ* & *FUJIWARA*

Claims 5, 10-13, and 22-25 were rejected under 35 U.S.C. § 103(a) as allegedly unpatentable in view of *Puz* and further in view of *Fujiwara*. This rejection was made in clear error.

In *KSR Int'l Co. v. Teleflex Inc.* (2007), the Supreme Court reiterated (citing *In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006)): "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."

The Final Office Action of July 21, 2010, erroneously asserted (pgs. 11-17) that *Puz* and *Fujiwara*, if combined, reasonably show or suggest the claimed tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, as

claimed. The Final Office Action also erroneously asserted (pgs. 11-17) that *Puz* and *Fujiwara*, if combined, reasonably show or suggest a database server that has access to a parameter value of the tag during execution of the database statement, as claimed. *Puz* and *Fujiwara* do not reasonably show or suggest at least these two features separately, let alone combined in the same way as recited in Claims 5, 10-13, and 22-25. Appellant respectfully submits that the rejection lacks the rational underpinning, based in fact, that is required by law to sustain the rejection under 35 U.S.C. § 103(a).

i. ***Puz* and *Fujiwara* do not reasonably show or suggest a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement.**

Not only do *Puz* and *Fujiwara* fail to reasonably show or suggest a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, but the claimed tag would not have been possible using the techniques of *Puz* and *Fujiwara*.

Puz relies on the position of a marker within the database statement because the marker is used as a placeholder for another database statement (pars. 33, 38). In other words, a security marker within a first database statement is replaced with a second database statement. The first database statement and the second database statement are combined to form a final database statement for execution by a database server (pars. 33, 38). The security marker in *Puz* must be embedded in the first database statement in order to hold a place within the first database statement. *Puz* could not replace the security marker with the second database statement to form

the final database statement if the security marker was not embedded in the first database statement.

Fujiwara does not fill the fundamental gaps left by *Puz*. *Fujiwara* (pars. 71-74) mentions that references to masked columns in an original database statement are replaced with corresponding masking function calls to form a translated database statement. *Fujiwara* explains (par. 73): "As can be seen the translation process 1210 is simply a textual replacement in the original query of the masked column references by their corresponding function calls." Thus, the masked column references serve as placeholders for the corresponding function calls.

Replacing a placeholder embedded in a database statement with text also embedded in the database statement, as mentioned in both *Puz* and *Fujiwara*, does not reasonably show or suggest the claimed tags, which are (a) included in a request to execute a database statement, and (b) not embedded in the database statement. In *Puz* and *Fujiwara*, the placeholder must be embedded in the database statement in order to hold a place within the database statement. Thus, the placeholder would not serve any purpose if the placeholder was not embedded in the database statement. Therefore, a tag that is (a) included in a request to execute a database statement, and (b) not embedded in the database statement, as claimed, would not have been possible using the teachings of *Puz* and *Fujiwara*.

Unlike the cited art, taken individually or combined, the recited tags are (a) included in a request to execute a database statement, and (b) not embedded in the database statement. The rejection lacks the rational underpinning, based in fact, that is required by law to sustain the rejection under 35 U.S.C. § 103(a). Therefore, the rejection of Claims 5, 10-13, and 22-25 cannot be sustained under 35 U.S.C. § 103(a). Appellant respectfully requests reversal of the

rejection of Claims 5, 10-13, and 22-25 under 35 U.S.C. § 103(a). Claims 5, 10-13, and 22-25 have been extensively examined and are believed to be in condition for allowance.

ii. *Puz* and *Fujiwara* do not reasonably show or suggest a database server that has access to a parameter value of the tag during execution of the database statement.

Not only do *Puz* and *Fujiwara* fail to reasonably show or suggest a database server that has access to a parameter value of the tag during execution of the database statement, as claimed, but the database servers in *Puz* and *Fujiwara* could not possibly have access to any parameter values of the security marker during execution of the database statement. Even if the security marker of *Puz* was analogous to the claimed tag, which it is not, *Puz* still fails to reasonably show or suggest a database server that has access to a parameter value of the tag during execution of the database statement.

In *Puz* (pars. 33, 38), the security marker, embedded in a first database statement, is replaced with a second database statement to form a final database statement prior to execution of the database statement. Thus, the database server in *Puz* does not receive the security marker and could not possibly have access to any parameter values of the security marker during execution of the database statement.

Fujiwara does not fill the fundamental gaps left by *Puz*. In *Fujiwara* (pars. 73-74), the references to masked columns in an original database statement are replaced with corresponding masking function calls to form a translated database statement. The database server in *Fujiwara* merely receives the translated database statement for execution. The database server in *Fujiwara* does not receive a tag or any other information other than the translated database statement for execution. Therefore, not only does *Fujiwara* fail to show a database server that has access to a

parameter value of the tag during execution of the database statement, as claimed, but the database server in *Puz* could not have possibly accessed any parameter values of the tag during execution of the database statement.

Unlike the cited art, taken individually or combined, Claims 5, 10-13, and 22-25 feature a database server that has access to a parameter value of the tag during execution of the database statement. The rejection lacks the rational underpinning, based in fact, that is required by law to sustain the rejection under 35 U.S.C. § 103(a). Therefore, the rejection of Claims 5, 10-13, and 22-25 cannot be sustained under 35 U.S.C. § 103(a). Appellant respectfully requests reversal of the rejection of Claims 5, 10-13, and 22-25 under 35 U.S.C. § 103(a). Claims 5, 10-13, and 22-25 have been extensively examined and are believed to be in condition for allowance.

iii. Conclusion regarding Claims 5, 10-13, and 22-25

As discussed above, at least two substantial features of Claims 5, 10-13, and 22-25 are completely absent from *Puz* and *Fujiwara*. Even if *Puz* and *Fujiwara* did reasonably show or suggest the two features separately, which they do not, the combination of *Puz* and *Fujiwara* do not reasonably show or suggest that, during execution of a database statement, a database server is provided with access to a parameter of a tag that is (a) included in a request to execute the database statement, and (b) not embedded in the database statement. Additionally, *Puz* and *Fujiwara* do not reasonably show or suggest a database server that has access, during execution, to information from a tag that does not conform to a database language of the database statement. Therefore, the rejection of Claims 5, 10-13, and 22-25 cannot be sustained under 35 U.S.C. § 103(a).

Due to the fundamental distinctions already identified, additional features recited in

Claims 5, 10-13, and 22-25 are not separately addressed in this Appeal Brief. Appellant

respectfully requests reversal of the rejection of Claims 5, 10-13, and 22-25 under 35 U.S.C. §

103(a). Claims 5, 10-13, and 22-25 have been extensively examined and are believed to be in

condition for allowance.

D.   CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the Examiner committed clear

error in rejecting Claims 1-26 under 35 U.S.C. § 112, first paragraph, Claims 1-4, 6-9, and 14-21

under 35 U.S.C. 35 U.S.C. § 102(e), and Claims 5, 10-13, and 22-25 under 35 U.S.C. 35 U.S.C.

§ 103(a). Appellants therefore respectfully request that the Honorable Board reverse the

rejection of Claims 1-26 under 35 U.S.C. § 112, first paragraph, Claims 1-4, 6-9, and 14-21

under 35 U.S.C. 35 U.S.C. § 102(e), and Claims 5, 10-13, and 22-25 under 35 U.S.C. 35 U.S.C.

§ 103(a). All pending claims have been extensively examined and are believed to be in

condition for allowance.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

/EricLSutton#61173/
Eric L. Sutton
Reg. No. 61,173

Date: December 30, 2010
2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone: (408) 414-1080 ext. 210
Facsimile: (408) 414-1076

VIII. CLAIMS APPENDIX

1. (Previously Presented) A machine-implemented method for executing a database

statement, the method comprising the steps of:

a database server receiving a request to execute the database statement, wherein the

request includes the database statement and a tag that does not conform to a

database language of said database statement, wherein said tag is not embedded in

said database statement;

wherein said tag specifies at least one parameter field and at least one parameter value;

in response to receiving the request, said database server storing information from the tag

in a manner that is associated with said database statement and accessible to a tag

access mechanism;

said database server executing said database statement, wherein during execution of said

database statement said database server provides access to one or more of the at

least one parameter value through said tag access mechanism provided by said

database server.

2. (Previously Presented) The method of claim 1, wherein the database statement is written

in a language in which results desired are specified by the database statement, and no

procedures for obtaining the results desired are specified by the database statement.

3. (Original) The method of claim 1, wherein a priority for executing the database statement

is determined based on the at least one parameter value.

4.   (Original) The method of claim 1, wherein a security level is associated with the at least

one parameter such that whether the database is entitled to access a component is based

on the at least one parameter.

5.   (Original) The method of claim 1, wherein the at least one parameter is accessible to a

systems administrator.

6.   (Original) The method of claim 1, wherein the at least one parameter is related to user

context information.

7.   (Original) The method of claim 1, wherein the tag comprises an indicator of a beginning

of the tag, and an indicator of an end of the tag.

8.   (Original) The method of claim 7, wherein the at least one parameter value is located

between the indicator of the beginning and the indicator of the end of the tag.

9.   (Original) The method of claim 8, wherein each of the at least one parameter fields

comprises an indicator of a beginning of the parameter field, followed by the parameter

value, which in turn is followed by an indicator of an end of the parameter field.

10.   (Previously Presented) The method of claim 1 wherein the at least one parameter value

can be accessed without accessing memory allocated to a database session, wherein the

database statement was issued within the database session.

11.   (Previously Presented) The method of claim 4, wherein the at least one parameter value

can be accessed without accessing memory allocated to a database session, wherein the

database statement was issued within the database session.

12.   (Previously Presented) The method of claim 2, wherein the at least one parameter value

can be accessed without accessing memory allocated to a database session, wherein the

database statement was issued within the database session.

13.   (Previously Presented) The method of claim 1, wherein the at least one parameter value

can be accessed after a session window has closed, wherein the database statement was

issued within the session window.

14.   (Previously Presented) A non-transitory machine-readable storage medium storing one or

more sequences of instructions, which when executed by one or more processors, causes

the one or more processors to perform a method comprising the steps of:

a database server receiving a request to execute a database statement, wherein the request

includes the database statement and a tag that does not conform to a database

language of said database statement, wherein said tag is not embedded in said

database statement;

wherein said tag specifies at least one parameter field and at least one parameter value;

in response to receiving the request, said database server storing information from the tag

in a manner that is associated with said database statement and accessible to a tag

access mechanism;

said database server executing said database statement, wherein during execution of said

database statement said database server provides access to one or more of the at

least one parameter value through said tag access mechanism provided by said

database server.

15. (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the database statement is written in a language in which results desired are specified by the database statement, but no procedures for obtaining the results desired are specified by the database statement.

16. (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein a priority for executing the database statement is determined based on the at least one parameter value.

17. (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the at least one parameter is accessible to a systems administrator.

18. (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the at least one parameter is related to user context information.

19. (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the tag comprises an indicator of a beginning of the tag, and an indicator of an end of the tag.

20. (Previously Presented) The non-transitory machine-readable storage medium of claim 19, wherein the at least one parameter value is located between the indicator of the beginning and the indicator of the end of the tag.

21. (Previously Presented) The non-transitory machine-readable storage medium of claim 20, wherein each of the at least one parameter fields comprises an indicator of a beginning of the parameter field, followed by the parameter value, which in turn is followed by an indicator of an end of the parameter field.

22.   (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the at least one parameter value can be accessed without accessing memory allocated to a database session, wherein the database statement was issued within the database session.

23.   (Previously Presented) The non-transitory machine-readable storage medium of claim 26, wherein the at least one parameter value can be accessed without accessing memory allocated to a database session, wherein the database statement was issued within the database session.

24.   (Previously Presented) The non-transitory machine-readable storage medium of claim 15, wherein the at least one parameter value can be accessed without accessing memory allocated to a database session, wherein the database statement was issued within the database session.

25.   (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein the at least one parameter value can be accessed after a session window has closed, wherein the database statement was issued within the session window.

26.   (Previously Presented) The non-transitory machine-readable storage medium of claim 14, wherein a security level is associated with the at least one parameter such that whether the database is entitled to access a component is based on the at least one parameter.

IX. EVIDENCE APPENDIX PAGE

None.

X. RELATED PROCEEDINGS APPENDIX PAGE

None.